BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

DETECTION OF ANOMALIES IN DATA CENTER NETWORK TRAFFIC
DETEKCE ANOMÁLIÍ V SÍŤOVÉM PROVOZU DATOVÉHO CENTRA

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR — **Leila Korzhasbayeva**
AUTOR PRÁCE

SUPERVISOR — **doc. Ing. Radim Burget, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Studentka: Leila Korzhasbayeva ***ID:*** 185930
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Detekce anomálií v síťovém provozu datového centra

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou analýzy časových řad a rozpoznáváním anomálií v jejich průběhu. Seznamte se s problematikou konvolučních hlubokých neuronových sítích, sítěmi LSTM a problematikou autoenkodérů. Navrhněte neuronovou síť a na vhodném případě z oblasti bezpečnosti, např. provozu serveru demonstrujte schopnost identifikace nestandardních událostí. Obdržené výsledky vyneste vhodně do grafů a tabulek a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[2] Karpathy, Andrej, et al. "Large-scale video classification with convolutional neural networks." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014.

Termín zadání: 1. 2. 2019 ***Termín odevzdání:*** 12. 8. 2019

Vedoucí práce: doc. Ing. Radim Burget, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRACT

In large companies there are plenty of critically important machines working around the clock every day. a simple Log Management solution is not always sufficient to capture all the data that flows through the production environment. Even a security analyst is not always able to track every resource in the environment, catching changes in normal traffic. The main motivation of the work is the automation of log analysis to facilitate the work of network and security operations center specialists. The work is solved in the context of improving and optimizing the performance of security analysts in cooperation with Log management and other monitoring tools that should serve as data sources. With the fact that most companies have already implemented Security Event and Information Management in production, there are still deviations in the running of the company, so-called anomalies. The solution presented below aims to discover similar, unconscious events in the data center network traffic. There are currently several directions in the solution: manual, semi-manual (set threshold), statistical, machine learning (advanced statistical). Described approach to the problem is backed up by methods of data analysis, especially on machine learning algorithms. Three machine learning algorithms were selected: isolation forest, support vector method, long-term memory elements model. Data was collected from production environments of different companies for separate time segments to test functionality on different types of data. In the first case, which was used in the semester work, the search for anomalies was only one value of the logs when the other data needed a summary evaluation. These are different types of problems: uni variate versus multivariate. The difference is the number of monitored values or indicators, in the case of this work it is more than a thousand monitored indicators for certain times. The final solution model produces output for several values, given as a prediction, yet evaluating whether there is an anomaly. The evaluation process is based on the difference between the test data and the model output, which is a prediction of all values at the next time points.

KEYWORDS

Data center network, machine learning, neural networks, network traffic, security

ABSTRAKT
Ve velkých společnostech existuje spousta kriticky důležitých strojů pracujících bez přestávky každý den. Jednoduché Log Management řešení není vždy dostatečné k zachycení všech dat, která tečou produkčním prostředí. Ani bezpečnostní analytik není vždy schopen sledovat každý zdroj v prostředí, chytat změny v běžném provozu. Hlavní motivací dané práce je automatizace analýzy logů pro usnadnění práce specialistům síťových a bezpečnostních operačních středisek. Práce je řešena v kontextu vylepšení a optimalizace výkonu bezpečnostních analytiků při součinnosti Log management a jiných monitorovácích nástrojů, které by měli sloužit jako zdroje dat. s tím, že v současné době většina firem již má v produkci implementováno Security Event and Information Management, stále se objevují odchylky v chodu společnosti, tak zvané anomálie. Řešení, předložené níže, má za cíl objevovat podobné předem nevědomé události v síťovém provozu datového centra.
V současnosti existuje několik směrů v řešení dané otázky: manuální, polo manuální (nastavený práh), statistický, pokročilý statistický (metody strojového učení). Popsaný návrh přístupu k problému je založen na metodách oblasti analýzy dat, zejména na algoritmech strojového učení.
Byly vybrány 3 algoritmy strojového učení: izolační les, metoda podpůrných vektorů a rekurentní neuronová sít s prvky dlouhé krátkodobé paměti. Data byla nasbírána z produkčních prostředí různých firem za oddělené segmenty času pro testování funkčnosti na různých typech dat. v prvním případě, který byl použit v semestrální práce, se jednalo o vyhledávání anomálií jenom v jedné hodnotě z logů, když ostatní data potřebovala sumární vyhodnocení. Jedná se o různé typy problémů: univariate versus multivariate. Rozdílem je počet sledovaných hodnot či indikátorů, v případě této práce se jedna o více než tisíc sledovaných ukazatelů pro určité časové okamžiky. To znamená, že se jedná o vícerozměrné časové řady.
Během psání této práce byly zkonstruovány tři modely pro detekci anomálií. První z nich byl vytvořen na základě klasické metody podpůrných vektorů pro jednu třídu, jejíž obecnou podstatou je oddělení datového vzorku od začátku souřadnice.
Druhý model použil metodu izolačního lesa, jehož obecná podstata je následující: je vytvořen určitý počet rozhodovacích stromů (každý strom se staví do vyčerpání datového vzorku) a pro každou novou větev stromu je vybrána náhodná proměnná a náhodná hodnota této proměnné, podle které k tomu větvení dojde. Poté se u každého objektu zvažuje míra jeho anomálie, která se rovná průměrné hodnotě hloubek listů stromů, do kterých tento objekt spadl. Nakonec, vypočítaná míra anomálií je porovnána s určitou hraniční hodnotou a na základě toho se rozhodne, zda objekt považovat za anomálii nebo ne.
Třetí model byl založen na neuronových sítích. Algoritmus pro detekci anomálií ve vícerozměrných časových řadách pomocí rekurentních neuronových sítí je následující: nejprve algoritmus predikuje hodnoty každého z pozorovaných indikátorů v čase t, a pak hodnoty těchto predikcí jsou porovnány se skutečnými hodnotami těchto indikátorů v čase t. Anomálie jsou ty objekty, u kterých rozdíl mezi předpovězenými hodnotami a skutečnými hodnotami překračuje určitou hranici.
Je třeba poznamenat, že problém odhalení anomálií patří do třídy úloh učení bez učitele (unsupervised learning), což znamená, že skutečné hodnoty cílové proměnné nejsou k dispozici. v důsledku toho je kvalita takového algoritmu často hodnocena odborně, a to i v této práci. Provedeným výzkumem bylo zjištěno, že pro mále množství dat, izolační les je nejefektivnější způsob řešení, zatímco rekurentní neuronové sítě jsou výkonnější pro vysoké množství vstupních dat. Nicméně, vytvořené řešení poskytuje možnosti nasázení na prostředí s malým a velkým tokem dat, přizpůsobitelná nastavení pro lepší výkon a ulehčení práce pro specialisty v oblasti bezpečnostního monitoringu.

DECLARATION

I declare that I have written the Bachelor's Thesis titled "Detekce anomálií v síťovém provozu datového centra" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Bachelor's Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

ACKNOWLEDGEMENT

Ráda bych poděkovala vedoucímu diplomové práce panu doc. Ing. Radimovi Burgetovi Ph.D. za velkou trpělivost a podnětné návrhy k práci. Kromě toho bych chtěla se omluvit za všechny způsobené nepříjemností.

Brno ....

author's signature

CONTENTS

INTRODUCTION

Nowadays computer technologies are completely gained a foothold in life of every person. Our devices know when we go to sleep, how many steps we take, what do we eat. Everything is being monitored by these small electronic things and people have to care about keeping it secure. Machines are even being learned how to think like a human. Science field that works in this area is called Machine Learning. Machine learning technology is usable in a vast area of applications, and more use cases are being found out as time passes by. It gives an opportunity to systems to boost the dynamic environments, where the input signals are unknown, and the best solutions are about to be made based only on historical experience. Moreover, modern IT infrastructures, based on data centers, are generating a tons of traffic everyday and as every company stores their assets in data centers, it should be monitored properly both physically and on virtual level. Virtual level could be defined as Device Health monitoring, Network monitoring and Monitoring of specific services or storages. Conventional resolutions monitor network traffic and determine attack activities from legitimate sources supported by applied mathematical divergence. And there are still Human Analysts have their places. Of course, all the algorithms have to be controlled before the production and afterwords, but who would check the possible human errors? Not every Security Analyst is capable to immediately detect an anomalous behavior of defined log sources, even using the SIEM or other monitoring tools, where the small network tests have their places. That is why there is an additional approach helping Security Analysts to process more data, find the previously unknown connections and react to an incident immediately. For now, machine learning techniques are known as a common approach for developing network anomaly detection models. But there are not only machine learning approaches could be used. One of the most interesting topics that got boosted and popularized really fast is an AI - Artificial Intelligence. In general, machine learning is the way of achieving the AI and, leading to the approach of this project, Deep Learning is the technique of implementing ML. The Deep Learning is set on Artificial Neural Network (ANN). The one of methodologies used is describing the usage of an LSTM type of ANN for the Anomaly Detection in Real Network Traffic. It is expected to be the favorite approach for the project, whereas the other models are going to be tested too. Within the given data, there were successfully detected anomalies by multiple models, as well as, one of the described models got the second place in the related competition of Junction Hackathon in Helsinki. The main problem for this paper is the detection of anomalies in multidimensional time series, while the previous work was solving only one dimensional anomaly detection, standing for detecting outliers only for one value of indicator. Here are native

solutions for anomaly detection are presented, including Isolation Forest, Support Vector Machine and, as mentioned above, RNN LSTM.

1 DATA CENTER NETWORK TRAFFIC

New goals of business and trend of technologies are making a big difference to how people understand a field of IT and it does give brand new challenges to infrastructures, that past data centers were not prepared for it. The goal of modern data center networks is to host multiple residents with different workloads, however these workloads are being developed much more faster, than the base of it - load handlers - are being modified to fulfill the request. One of the most simple network services can be defined as responses to calls of API functions. But nowadays they do operate with much more pressure. Even individual computers are running several business loads, less consumer and entertainment ones, but, there is still a need for data center networks to handle a really huge amount of request efficiently. That requests consist of network traffic produced by mobile devices, application abound, usual workstations. The Internet of Things will even multiple the number of devices generating traffic on networks. Consequently, IP traffic is increasing in big steps. Touching business, almost 80As it has a big impact to business, proper security monitoring should be established. As on physical layer, so on virtual layer as well. Prevention of attacks is one of the domains, where there is a possible prediction is present. To get better picture of the problem, a deep investigation in root domain is required. This chapter is all about data center definition, networking and proper monitoring.

1.1 Data center denotation

According to Technopedia[1], a data center "is a repository that houses computing facilities like servers, routers, switches and firewalls, as well as supporting components like backup equipment, fire suppression and air conditioning". From this descriptions it is obvious, that data center is a physical place full of racks with specific hardware (servers, routers, firewalls, switches, etc.) and require a bunch of assistant systems such as air conditioning and climate control, fire and smoke detection, water damage prevention, identification and access control. Moreover, it is the base of every IT infrastructure. Most of the time, data centers are being understood as whole unit, yet they consists of some technical components, a specific hardware, as was mentioned above (servers, routers, firewalls, switches, etc.). This whole description could be defined with term "data center hardware". The most of them are being divided to several categories:

- Storage resources: All data has to be stored properly based on internal or national rules. Storage systems are made to save valuable assets of enterprise.

[1] Technopedia

Hard drives, tapes are the most common carriers.

- Computing components: Drivers of applications. These are servers to give a customer requested computing ability.
- Network equipment: Devices that provide secure connectivity inside the data center and outside of it, such connection to internet: routers, switches,firewalls, cables.
- Facility stuff: This category includes all other supplies supporting the normal functioning of data center. Racks, electricity supplies including uninterrupted power supply system, cooling systems, fire protection are all the things to assure the data center's work.

There is also physical security on place. It ensures the identification and access control to the facility.

1.2 Network Traffic

The Data center Network is playing a big role in functioning of every data center. It's main task is to interconnect every node in the architecture together. There are a bunch of possible physical topologies that could be implemented based on business needs: Fat Tree, JellyFish, DCell, Multi-tier leaf-spine, etc. Every of it has its own pros and cons and own requirements. But in this section traffic itself will be discussed, as an approach is getting network data for the input. Network traffic can vary from environment to environment based on applications it run and hardware resources, as it then makes difference to flow sizes and durations. It is obvious from a simple example: web search services are generating much less network flows than an average computing job. Basically, network traffic is data of some amount moving within the network at specified time point. These data are being wrapped to network packets, that loads the network. Proper monitoring and analysis of situation on network side is assuring Security in organization , because unusual network behavior is a possible sign of an attack. In field of data centers there are mostly two types of network traffic: east-west or north-south. It is could by imaginary moves from side to side: left-right or up-down. North-south is client-server communication prototype. It moves between data center and some defined location outside of it. In opposite side, east-west network traffic is describing packets traveling between nodes within one data center environment. There is one more term to be added a network flow. Network flow is "a sequence of packets sent from a particular source to a particular unicast, anycast, or multicast destination that the source desires to label as a flow. A flow could consist of all packets in a specific transport connection or a media stream. However, a flow is not necessarily 1:1 mapped to a transport connection.", as specified in RFC 3697[21].

1.3 Network Traffic Monitoring

As many data center hosts are providing Cloud services, the performance and reliability should be presented at it's best. This state could be achieved with non-stop processes, which means high-availability, security and monitoring have to be the part of the service. Monitoring matters because business is highly dependent on assets that are stored, processed and going through the data center and if something fails within the network, it causes major impact to business continuity and will cost resources, time and money. That's why monitoring is an amazingly strong way to deeply understand what problems or issues IT environment is facing right away. Common example of monitoring system functionality is sending HTTP requests to web service's fetch page to check the availability, or sending ICMP echo requests for the host availability check. Network monitoring systems are built to detect problems in connectivity, availability and load, while intrusion detection system monitors network for known threats, mostly, from the outside of environment - internet. There are a lot of solutions for Network monitoring giving a customer all perfectly shaped dashboards with schemas, such as Nagios, Solarwinds's Network Performance Monitor, Zabbix and much more not including the out-of-the-box solutions from hardware vendors, for example Checkpoint's SmartConsole. And all these solutions are being generating and fed by logs from all across the environment to be monitored. Logs are continuously recorded, time stamped, events, that are being generated by sources: hardware or software. It is used for deeper investigation in cases of breach or malfunctioning. Logs can be structured or unstructured and have many formats, such as NCSA Common log format, Windows events, syslog, Cron and many others.

```
1 127.0.0.1 user-identifier frank [10/Oct/2000:13:55:36 -0700] "GET /
    apache_pb.gif HTTP/1.0" 200 2326
```

Listing 1.1: NCSA Common Log Format

Despite having widely known format, all logs are created by developers for debugging the resource being logged, which means logs are vary depending on developers and systems. To get a wider picture Log management solutions were raised. It helps to maintain a big amount of logs. The functions of Log Management are end-to-end log collection, aggregation, analysis, reporting, storage and retention. It also required as must have solution by ISO 27001[2]: part A.10.10 is describing the requirements in detail. Main features of Log Management:

- Analyzing Logs for Relevant Security Intelligence
- Centralizing Log Collection

[2]ISO/IEC 27001 is an information security standard, part of the ISO/IEC 27000 family of standards

- Meeting IT Compliance Requirements
- Conducting Effective Root Cause Analysis
- Making Log Data More Meaningful
- Tracking Suspicious User Behavior

There is also another solution besides Log Management - SIEM. Security Information and Event Management is a term describing service that unites Security Information Management and Security Event Management. Security Information Management provides long-term storage, analysis and log data reporting, while Security Event Manager works with monitoring in real time, events correlation, dashboard view and notifications to analysts. Security Information and Event Management collects and manages all the defined logs as the Log Management does, but there is on big difference in between these two types. Security Information and Event Management simply has the correlation ability. Correlation increases the network security by simultaneously processing tons of events to detect unusual ones on the network. Real-time event correlation is all about proactively dealing with threats.

2 TIME SERIES ANOMALY DETECTION

2.1 Anomaly definition

Anomalies or Outliers are the observations that are somehow differ from the expected behavior. Following figures are explaining the concept of an anomaly: fig. 2.2 is showing point anomalies in Active Directory logs and fig. 2.1 is showing outliers - the samples, lying out of normal observations.

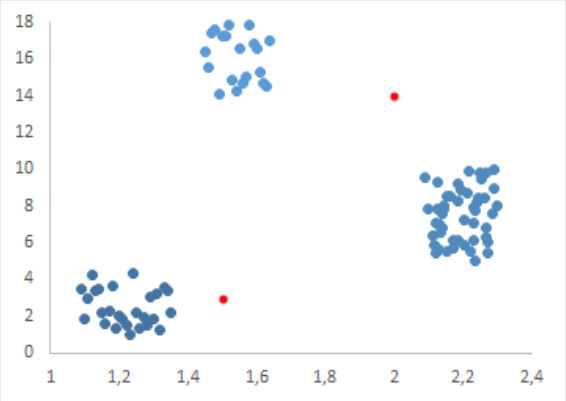

Fig. 2.1: Outliers

2.1.1 Types of Anomalies

There are three types of anomalies that are stated in studied literature: point anomalies, sequential anomalies and contextual anomalies.

- Point Anomalies are the samples of data that being classified as anomalous in respect to the rest of data. This is the simplest type of anomaly and is the focus of majority of research on anomaly detection.[1] Point anomalies in fig. 2.2 and fig. 2.1 are marked as red.
- Sequential anomalies are the sequences of points in data that are anomalous in respect to the rest of data. Points in defined anomalous sequences are not anomalous themselves. A high-level example: let's assume we have an event processor, that forwards data from dedicated environment to SIEM [1]; when one of the servers that is sending logs to event processor is disconnected before the expected time, from this point, event processor is forwarding constant number of messages about log source disconnection, that are differing from the usual traffic being forwarded out of that server. So, the sequence of these messages will be anomalous, but the messages themselves are not.

[1]Security Information and Event Management

- Contextual anomalies are the points or a sequence of points that are classified as anomalous based on the location of their neighbors. As an example we will analyze the internet banking usage. During the day, the summary of activities against the service is pretty high: starting from 6 am with 1000 activity observations and having its continuous peak at the midday with 5000 activity observations; we assume, that the normal amount of observations per time entity during the night hours is 100. When the monitoring will show you 1000 activity observations per time entry at 3 am, this will be considered as Contextual anomaly for the night traffic, but same amount will be fine for morning-day activity rates.

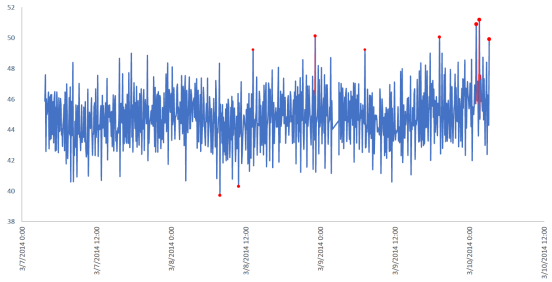

Fig. 2.2: Point anomalies

2.1.2 Application of Anomaly Detection

There are a lot of ways to apply anomaly detection to different topics. Description of several entries are follow:

- Medicine: unusual or unexpected symptoms of test results can be obtained to detect potential problems with health; the classification of the anomalous nature may depend on other factors such as age, gender, etc.
- Sports: In many sports there are a variety of parameters being recorded of players to evaluate the overall performance of one; these values can be shown as abnormal only in depending on a special subset of the recorded parameters.
- Fraud detection: the usual pattern of credit card transaction changes when the card has been stolen.
- Measurement errors detection: data gained from sensors may have measurement abnormal observations; analysts could proactively start the investigation of such failures.

2.2 Time Series

As the idea of the project is in detecting anomalies in network traffic, time series analysis took a meaningful part of data understanding. As it is understandable from the term itself, time series are defined as sequentially measured data over some (often equal) periods of time.

2.2.1 Main problems of Time Series

There are two main objectives for time series field: Time Series Analysis - determining the nature of the series, and Forecasting - predicting future values of the time series from present and past values. Both of these objectives require that the series model to be identified and, more or less, formally described. Once the model is defined, you can use it to interpret the data you have, for example, use prepared model to understand the seasonal change in prices of products. Ignoring the depth of understanding and the validity of the theory, you can then extrapolate the series based on the model found, i.e. predict its future values. The concept of time series analysis is used to separate this task first of all from simpler data analysis tasks, when there is no natural order of receipt of observations, and, secondly, from spatial data analysis, in which observations are often associated with geographical location. The time series model in a general sense reflects the idea that close observations in time will be more closely related than remote ones. In addition, time series models often use an unidirectional order in time in the sense that values in a series are expressed in some form in terms of past values, and not in subsequent ones. Based on this division to two types of time series problems, there are special sub tasks added. Time series Analysis field solves Trend Education, Seasonal components search, anomaly detection and segmentation. While Forecasting has following basic techniques: Autoregressive (AR), Moving Average (MA), Adaptive, it might grow to some ascended mixes, such as ARIMA, SARIMA and a lot more.

2.2.2 Components

Most of the regular components of time series belong to two classes: they are either a trend or a seasonal component. A trend is a general systematic linear or non-linear component that may change over the time. The seasonal component is a periodically repeating component. Both of these kinds of regular components are often present in the series at the same time. For example, company sales may increase from year to year, but they also contain a seasonal component. For example, as a rule, 25% of annual sales fall in December and only 4% in August. This general model can be understood based on "classic" series - Series G, representing monthly international air

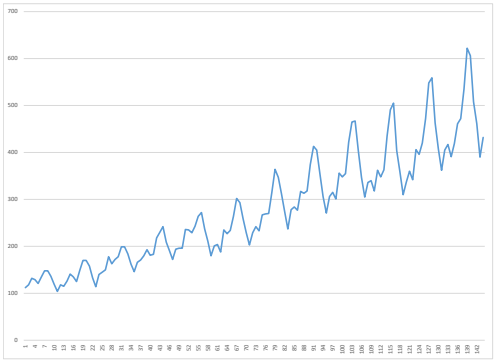

Fig. 2.3: G Series, monthly representation of international air travel in thousands

travel in thousands for 12 years from 1949 to 1960. The monthly traffic chart clearly shows an almost linear trend i.e. there is a steady increase in traffic from year to year - approximately 4 times more passengers transported in 1960 than in 1949.[2] At the same time, the nature of monthly transportation is repeated, they have almost the same character in each annual period - there are more transportation during vacation periods than in other months. This example shows a fairly specific type of time series model in which the amplitude of seasonal variations increases along with the trend. Such models are called multiplicative(below defined) seasonality models. But there are two more components making difference to analysis: cyclic and irregular (random). Cyclic component reflects cyclical changes in time series levels for periods over 1 year. This component is associated with business cycles; its frequency is from 2 to 10 years. The cyclic component is difficult to identify if analyzing data for a short, relatively cycle time period. In this case, the cyclic component cannot be separated from the trend component. An irregular component reflects irregular fluctuation levels of the time series, which cannot be predicted, are the result of one-time rather than systematic events affecting the levels of the series. There are two main ways (models) by which time series components can interact:

1. additive model: $y_i = T_i + C_i + S_i + I_i$;
2. multiplicative model: $y_i = T_i * C_i * S_i * I_i$,

where y_i is the level of a number of dynamics; T_i is the trend component; C_i is a cyclic component; S_i is the seasonal component; I_i is an irregular component.

The choice between the additive and multiplicative model depends on the nature of the source data. For example, if each year the amplitude of cyclical and seasonal changes is constant, use an additive model; if the amplitude of these changes increases with an increase in the indicator, use a multiplicative model. In the practice of forecasting a multiplicative model is used more often.

2.3 Detection methods

In this section common anomaly detection techniques will be described. Different points are divided by the learning types — the way how the model is being fed with data. In general, there are two kinds of model guidance to detect anomalies: supervised and unsupervised.

2.3.1 Supervised

Supervised learning is a method that gets the knowledge from labeled dataset. Labeled data are providing an answer key that model can use for the evaluation of its accuracy on training data. So, that means that a labeled data set of animal photos will tell the algorithm on which photos were tigers, zebras and humans. When the model gets an input, it compares the sample with ones it had in training examples to add the right label to the input sample. There are a lot of areas, where the supervised learning could be used, but for classification and regression problems it is used with a priority. Classification problems goals are to identify the input data as the part of a particular class. The algorithm is fed by pre-labeled data set and at the cross validation part it is evaluated by how accurately it can classify the new data correctly. Regression problems are all about continuous data. A linear regression is about expecting one unknown value depending on another one, which is known. In real cases it could sound like prediction of the price for an apartment in a city center based on location, square footage and transportation available. The Supervised algorithms are the best suited to challenges, where there are a lot of references available with which to fit the model. We can add following algorithms as Supervised learning:

- Decision trees
- Supported Vector Machine
- k-nearest neighbors
- Bayesian networks
- Supervised Neural Network

2.3.2 Unsupervised

If there is no training data available the Unsupervised techniques are required. The data are coming without any instructions, means there is no specific expected output or a correct answer. So the model has to analyze the data structure itself and try to find useful features. Depending on problem, model can outcome in different ways:

- Clustering
- Anomaly detection

- Assoviation
- Autoencoders

Unsupervised algorithms tend to find previously unknown patterns in data. But there is no way to determine an accuracy without having data to compare. It is also could be useful to uncover data for further usage with supervised models.

For the anomaly detection problem, there are several methods exist, which will be described and implemented later on:

- k-mean
- Isolation Forest
- Local Outlier Factor
- Markov Chains
- OneClass SVM
- RNN

2.4 Possible solution

For the goal of this project the Unsupervised solution was chosen. It has the big advantage of supervised learning in that you do not need any pre-labeled data. Another one advantage is much easier adaptation for the streaming data. The main model to be implemented is based on Recurrent Neural Networks, but the other possible solutions were tested as well. Between other solutions tested Isolation Forest and OneClass SVM took a place.

2.4.1 Isolation Forest

The idea of an isolating forest[12] is based on the Monte Carlo principle: a random partition of the feature space is carried out, such that in the average isolated points are cut off from normal, clustered data. The final result is averaged over several starts of the stochastic algorithm. The isolation tree algorithm is to construct a random binary decision tree. The root of the tree is the entire space of attributes; in the next node, a random attribute and a random partitioning threshold are selected, sampled from a uniform distribution over the interval from the minimum to the maximum value of the selected attribute. The criterion of the stop is the identical coincidence of all objects in the node, that is, the decision tree is completely built. The answer in the sheet, which also corresponds to the anomaly score algorithm, is the depth of the sheet in the constructed tree. It is argued that it is typical for anomalous points to appear in leaves with a shallow depth, that is, in leaves close to the root, when in order to partition normal data with hyperplanes of the cluster, the tree will need to build another several levels. Moreover, the number of such levels is proportional

to the cluster size; consequently, anomaly score is proportional to the points lying in it. This means that objects from small clusters, which are potentially anomalies, will have anomaly score lower than from normal data clusters. The algorithm has a number of significant advantages:

- The algorithm recognizes anomalies of various types: both isolated points with a low local density, and clusters of anomalies of small sizes.
- The complexity of the isolation tree is – $O(nlogn)$, which is more efficient than most other algorithms.
- It does not require significant memory costs, unlike, for example, metric methods, which often require the construction of a matrix of pairwise distances.
- There are no parameters requiring selection.
- Invariant to scaling features; does not require setting metrics or other a priori information about the data device.
- Resistant to the curse of dimensionality.

2.4.2 One Class Support Vector Machine

The support vector method is used to search for anomalies in systems where normal behavior is represented by only one class. This method determines the boundary of the region in which the normal data instances are located. For each studied specimen, it is determined whether it is located in a particular region. If an instance is outside the region, it is defined as abnormal. In general, the main idea of the algorithm (in the case of classification) is to separate classes with a hyperplane so as to maximize the distance (gap) between them. Initially, the algorithm was able to work only with linearly separable classes, but in the 90s of the last century, the method became especially popular due to the introduction of "Kernel Trick"(1992), which allowed working effectively with linearly inseparable data. The kernel is a function that is able to transform a feature space (including non-linearly), without directly converting features. It is extremely effective in terms of calculation and potentially allows you to get infinite-dimensional feature spaces. The idea is that classes that are linearly inseparable in the current attribute space can become separable in spaces of higher dimension. One Class SVM is one of the forms of the classical algorithm, however, as the name implies, for its training we need only have one class. If we are dealing with the novelty detection task, where only "good"observations without anomalies are available for training, we can use this model and learn how to say for each new observation whether it is anomalous or not. The general idea is to transform the attribute space and draw a dividing hyperplane so that the observations lie as far away as possible from the origin. As a result, we get a border on one side of which the observations from our pure training set are packed as densely as possible, and

on the other side abnormal values will be found, not similar to what the algorithm saw during training. There are few cons while working with SVM:

- It can overfit very much and produce a large number of false negative results if the separation gap is too small.
- It is mandatory to be absolutely sure that the training data does not contain any outliers, otherwise the algorithm will consider them as normal observations.

But there are two pro's to balance:

- Thanks to kernel trick, the model is able to draw nonlinear dividing boundaries.
- It is especially convenient to use when there are not enough "bad" observations in the data to use the standard approach of learning with a teacher - binary classification.

2.4.3 Local Ourlier Factor

A more subtle problem of metric methods is the fact that all the assumptions underlying them are valid only in complementing each other: for example, the local density of a point lying in the center of a small cluster of anomalies can be higher than for any point from a large cluster of abnormal data. The opposite is also possible: an isolated anomaly point can be located, for example, in the center of mass of a cluster of normals, and then the average distance from it to its neighbors will be less than for normal points. it Exactly this property of metric algorithms the LOF (Local Outlier Factor) algorithm is trying to take into account. The intuition of the algorithm's formula is to compare the average reach of a point and its nearest neighbors. For representatives of normal data, it is true not only that the local density estimate is small, but that it is slightly different from the same estimate for the nearest neighbors.

3 NEURAL NETWORKS

Neural Networks are mathematical models inspired by biological neutral networks that form animal brains. These models are able to complete tasks that normal brain can do, such as recognition, vision, learning. As well as in animal brain, Neural Networks consist of small connected units called neurons. The function describing neuron (fig. 3.1) gives an output based on the inputs. Each connection between artificial neurons transmits signal to another one. There is a term for the connection itself – synapse. The postsynapse – receiving neuron – processes the signal, generates an output, which is then transmitted to other neurons. Commonly, the synapse is a real number that can be characterized with certain weight varying as learning proceeds, and the output value is being calculated by a non-linear function of the sum of all inputs. In case the weighted sum is zero, bias, sometimes can be understood as threshold, is added to make a non-zero output or to scale up the system response. Bias has the weight and input always equal to 1. The only difference between bias and threshold, that in case the threshold is set to another value than 0, the weights will just adapt themselves to adjust equation, i.e., weights (including bias) will absorb the threshold effects.

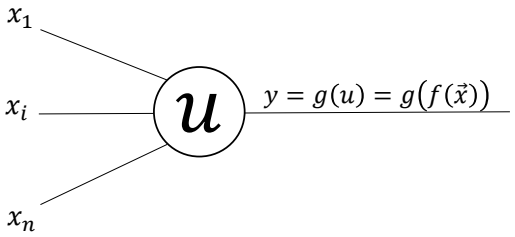

Fig. 3.1: Artificial Neuron

A set of input neurons consists of the neurons which are the first ones in any complete path in the graph. All input neurons have exactly one input and all of them together represents a problem to be solved by the neural network. All output neurons represent a possible solution of the problem on the input. For neural network to work it needs to be fed by he data into the input layer. Signals are being transferred from the input layer to the output layer all through the hidden ones as in fig. 3.2. Depending on how data are being transmitted, there are few Architectures of artificial neural networks. But the most ordinary neural network is the perceptron, which consists of one fully functional neuron. Let's take a look on fig. 3.1: if the output will be added to the right side instead of continuing the path to the possible

another neuron, the perceptron will be shown. Its output in algebraic way:

$$\text{output} = \begin{cases} 1 & \text{if } \sum_j w_j x_j \leq \text{threshold}, \\ 0 & \text{if } \sum_j w_j x_j > \text{threshold}. \end{cases} \tag{3.1}$$

Neural networks that consist of only one perceptron are very effective at their intended tasks and they can be combined into a multilayer perceptron (MLP) network to solve more difficult problems. But, as we saw before, perceptron can give us the output contains only of 0 or 1. The problem is that a small change in the weights or bias of any single perceptron in our network can cause the result to change, i.e., from 0 to 1. To easily overcome this kind of mistakes we use activation functions, sigmoid in this case.

Sigmoid function has an easily calculated derivative, which is important in calculating weights in the network, and easy to saturate. Inputs in sigmoid activated neuron can take on any values between 0 and 1 that will get more accurate output. However, recent works with neural networks, mostly deep learning models, have shown that rectifier activation function is more effective than sigmoid ones. It's gained by computational effectiveness: going back and forth through rectified linear units (ReLU) is just a simple if statement, while sigmoid requires to calculate the exponent. This difference is huge when dealing with multilayered network containing big amount of neurons in each.

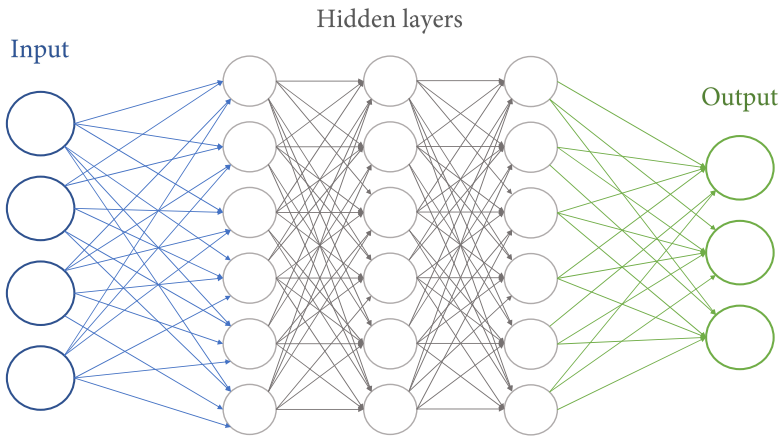

Fig. 3.2: Artificial Neural Network. Each node represents an artificial neuron and an arrow represents a connection from the output of the neuron to the input of another.

3.1 Architectures of artificial neural networks

In this section several types of neural networks that are related to the topic will be covered.

3.1.1 Feedforward

Neural networks, that were mentioned previously, where the output from one layer is used for an input of the next layer are called feedforward neural networks. This type of neural networks was the first and the simplest one. In model of feedforward network data are being transmitted without loops or cycles – information is always fed forward and never back. There are few models of feedforward networks.

Autoencoder is similar to the multilayer perceptron, but the output layer has exact the same number of neurons as the input layer. The purpose of autoencoder is to learn a representation for a set of data, i.e., reduce the amount of random variables by obtaining a set of principal variables – feature selection and feature extraction.

Convolutional neural networks (CNN) use a variation of MPL designed to require minimal preprocessing. It has shown good effectivity in analyzing visual imagery. The model is made of a recursive application of convolution and pooling layers, followed by inner product layers at the end of the network. A convolution layer is a linear transformation that preserves spatial information in the input image. Pooling layers simply take the output of a convolution layer and reduce its dimensionality (by taking the maximum of each (2, 2) block of pixels for example).[7]

However, there are other models of artificial neural networks in which feedback loops are possible. This models are called Recurrent Neural Networks.

3.1.2 Recurrent Neural Networks

Recurrent Neural Networks (RNN) transmit data in both ways: forward and backwards, which makes dynamic temporal behavior and allow information to persist. Unlike feedforward networks recurrent can use their own memory to process the input data. RNNs can be thought as multiple copies of the same network, each passing a message to a successor. Elements of the recurrent network are depicted as ordinary neurons with an additional cyclic arrow, which demonstrates that in addition to the input signal, the neuron also uses its additional hidden state. If you "deploy"such an image (fig. 3.3), you will get a whole chain of identical neurons, each of which receives its element of the sequence, enters a prediction and passes it along the chain as a kind of memory cell. It is necessary to understand that this is an abstraction, since it is one and the same neuron that works several times in a row. Simulation of memory in a

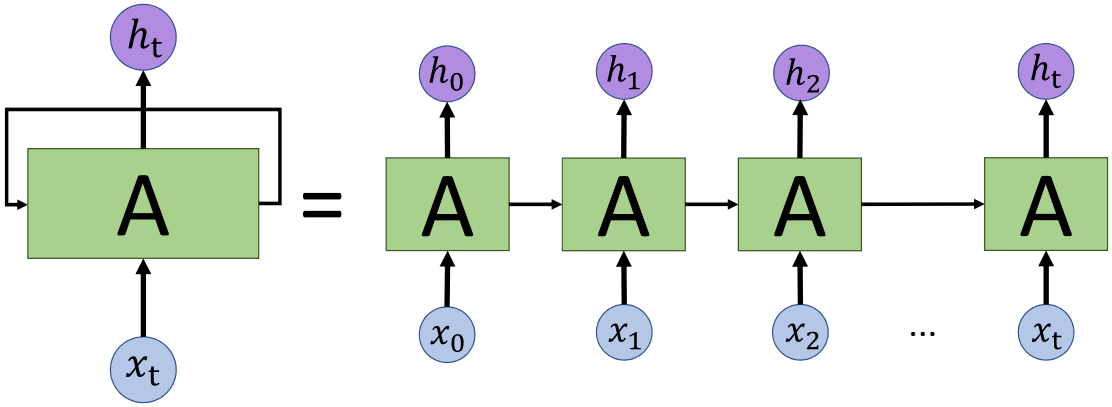

Fig. 3.3: A chunk of neural network, A, looks at some input x_t and outputs a value h_t.

neural network similarly introduces a new dimension in the description of the process of its work - time. Let the neural network receive a sequence of data, for example, text word by word or word-by-letter. Then each next element of this sequence enters the neuron at a new conditioned time point. By this time, the neuron already has accumulated experience since the beginning of information.

The main difference between all the types of recurrent neurons lies in how the memory cell inside them is processed. The traditional approach implies the addition of two vectors (signal and memory), followed by the calculation of the activation function of the sum.

But the memory realized in this way is very short. Since each time information in memory is mixed with information in a new signal, after 5-7 iterations the information is completely overwritten. Returning to the task of predicting the last word in the sentence, it should be noted that within a single sentence such a network will work well, but if it comes to a longer text, then the patterns in its beginning will no longer make any contribution to the network solutions near the end text, as well as an error on the first elements of the sequences in the learning process, ceases to contribute to the overall network error. This is a very conditional description of this phenomenon, in fact it is a fundamental problem of neural networks, which is called the vanishing gradient problem, and because of it the third "winter"of deep training at the end of the 20th century, when neural networks on one and a half Decades lost the leadership to machines of support vectors and boosting algorithms.

To overcome this disadvantage, the LSTM-RNN network (Long Short-Term Memory Recurrent Neural Network) was invented, in which additional internal transformations were added that operate the memory more cautiously.

Long Short-Term Memory Recurrent Neural Network

Long Short-Term Memory Recurrent Neural Network – a special kind of architecture of recurrent neural networks, capable of learning long-term dependencies. They were presented by Sepp Hochreiter and Jürgen Schmidhuber in 1997, and then refined and popularized in works of many other researchers. They perfectly solve a number of different tasks and are now widely used.

The LSTM network is an artificial neural network containing LSTM modules in place of or in addition to other network modules. The LSTM module is a recurrent network module that can store values for both short and long periods of time. The key to this feature is that the LSTM-module does not use the activation function inside its recurrent components. Thus, the stored value is not blurred over time, and the gradient will not vanish when using the Backpropagation through time method during network training.[26]

LSTM modules are often grouped into "blocks"containing different LSTM modules. Such a device is typical for "deep"multi-layer neural networks and facilitates the implementation of parallel computing with the use of appropriate equipment. The key to LSTMs is the cell state, the horizontal line running through the top of the fig. 3.4.

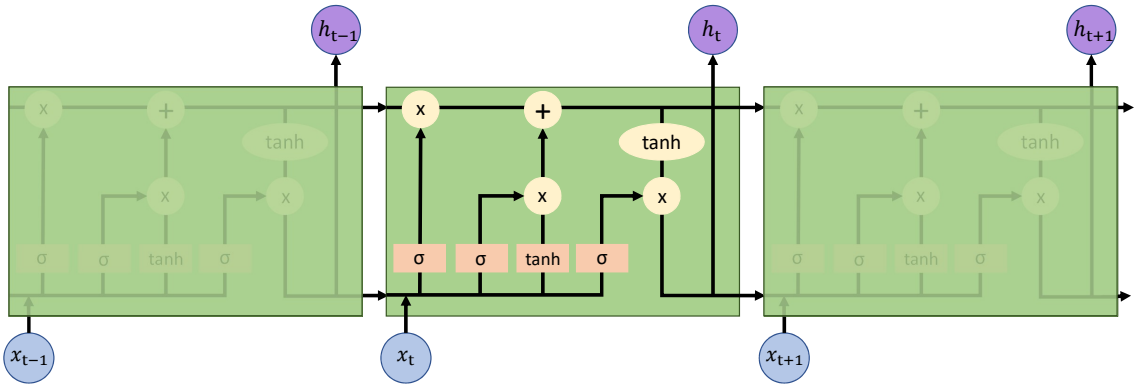

Fig. 3.4: The yellow circles – pointwise operations, like vector addition, orange boxes – learned neural network layers. Lines merging denote concatenation, line forking denote its content being copied (copies are going to different locations).

The cell status is like a conveyor row. It goes all the way down within the entire chain, with only some minor linear interactions. It is very easy for data to just flow along it without being changed. After all, the LSTM can delete this information aquired from the cell state: this process is managed by structures called filters or gates.

LSTM-blocks contain three "gates which are used to control the flow of information at the inputs and outputs of the memory of these blocks. These gates are implemented as a logistic function for calculating a value in the range [0; 1]. Multiplication by this value is used to partially allow or block the flow of information inside and out of memory. For example, the "input"gate controls the measure of the occurrence of a new value in memory, and the "forget fate"controls the measure of storing the value in memory. The "output "one controls the range to which the value stored in the memory is used to calculate the output activation function for the block. In some implementations, the input gate and the forget gate are embodied as a single gate. The idea is that the old value should be forgotten when a new value worthy of memorization appears.

3.2 Learning

The one more aspect of the model setting is learning. The learning problem is mostly formulized around the minimization of loss function. This function consists of an error and regularization fields. The first one evaluates how the data set is being fitted to a model and the second one is used in terms of overfitting prevention, which is ensured by the controlling of the effective complexity.

3.2.1 Hebbian learning

From the far forties of previous century D. O. Hebb created a learning hypithesis that was based on neural plasticity, which lately was called Hebbian Learning. This algorithm is an Unsupervised. The state itself is follows [10]:

> When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place on one or both cells such that A's efficiency as one of the cells firing B, is increased.

In a more familiar way it can be rephrased s the Hebbian Learning rule:

> If two neurons are on each side of a synapse and are activated simultaneously, then the weight and, consequently, bias of that synapse is increased.

The bias update can be represented as

$$\Delta w_{ij} = \eta * out_j * in_i \tag{3.2}$$

There is also a modification of the rule is available. The original was supplemented with this modification because the random coincidences will create compound strengths, and all the weights will tend to increase infinitely. The supplement is:

If two neurons are on each side of a synapse and are activated asynchronously, then the weight and, consequently, bias of that synapse is weakened or eliminated.

3.2.2 Backpropagation

The algorithm was introduced 1970-s and the importance of it is very high till now. At the backend of the algorithm, calculating a gradient for the computation of weights in model. The full name of the algorithm is explaining for itself: "the backward propagation of errors", while the error is being computed at the output and distributed backwards throughout the model layers[6]. The Backpropagation algorithm is highly useful for deep neural networks, where there is a tend to cause errors: image and sound recognition are good examples.

3.2.3 Algorithms

Here are the most important algorithms for training are described.

Gradient descent

Gradient (steepest) descent is the simplest training algorithm. It is the method of the first order, consequently, at the beginning it only requires an information from a gradient vector. Gradient descent requires many iterations for narrow, but long structures. It is recommended for big neural networks with a lot of parameters, because it only stores the gradient vector.

Newton's method

Newton's method has the second order, because it uses Hessian matrix. It finds the best training direction by using second derivatives of the loss function.[16]. Newton's method requires much less steps to find the minimum of the loss function, but the computational terms are quite expensive, because of the difficulty in Hessian matrix evaluation.

Conjugate gradient

This method is kind of an intermediate step between Newton's and Gradient descent methods. It avoids the information storage and processing required for the Hessian matrix and accelerates convergence from gradient descent. It is gained by searching in conjugate directions.

Levenberg-Marquart algorithm

This algorithm is known as the damped least-squares method, because it was designed to work with loss functions in the form of a sum of squared errors. It works with gradient vector and the Jacobian matrix. In general, from the beginning is starts in the gradient descent direction, but if it fails, the damping parameter increases. Otherwise, algorithm goes by a Newton's method. The disadvantage of this error is that it can not be applied to functions such as mean squared error or entropy error. As the conclusion: it requires a lot of memory, but very fast.

4 DESIGN AND IMPLEMENTATION

In this project the task of developing fully functioning IDS conceptual module is limited, instead, the module for Security Information and Event Management or Log management connection was created. As it was shown before, the data can be provided in a clear form, in the form of a set of functions that module can use in the input to the solution - the full pipeline has been applied. It is desirable to implement a multi-class method for presenting detector results at the end of the project, in order not only to detect the presence of anomalous activity, but also to determine its type, but for the sake of simplicity, a two-class detector has been implemented. All anomaly detectors, regardless of the models used in their design, are aligned with the principles of comparing reference data with the current situation and alarms. Any model from a statistical to an artificial intelligence system should have such a standard. In the case of a neural network, the standard will be created in the learning process during normal operation of the data network, after which the network functions, detecting deviations beyond the established limits. The complexity of this approach lies in the fact that:

- The network must be trained in "normal activity".
- It is also impossible to stop the data center or disconnect it from external networks.
- The explanation of the data, and, consequently, the period of study, must be sufficiently large: the functioning of the data center will almost guaranteed to differ in different periods.
- The rules of writing, describing the functioning of the data center in the usual mode, but here it is not applicable).

In this chapter provided theory is used to design a model that will be able to detect an anomaly. Data pre-processing is required to transform the data into a format usable by machine learning algorithms. Unsupervised learning is used to identify and learn patterns of network activity, as by default there has to be no human interaction in the pipeline. In a summary, there were three models built that are considered as native approaches for anomaly detection problem. One of them is Isolation Forest, the general essence of which is as follows: a certain number of decision trees is built (each tree is built until the selection is exhausted), and for each new branch of the tree, a random attribute and a random value of this attribute are selected, according to which it will happen branching. After that, for each object, the measure of its "anomaly"is considered, that is, the average value of the depths of the leaves of the trees into which this object fell. At the end, the calculated measure of "anomalies"is compared with a certain threshold and on the basis of this, a decision is made whether to consider the object as an anomaly or not. Another method is

One Class Support Vector Machine, and it uses the classical method of support vectors for one class, the general essence of which is to separate the sample from the origin. Long Short-Term Memory Recurrent Neural Network model has been chosen because when trained correctly, LSTM RNNs have the ability to impregnate themselves with the behavior of a training set. Intuitively meaning that when given a certain input samples,they have the ability to remember the context of the value of the samples, and to predict a coherent output in agreement with the context of the sample. The whole system is designed with thought that detecting is closely connected with prediction.

4.1 Data set definition

For this problem were three data sets obtained. First one, that was used for experiments in semestral project, consisted of uni variate table. So, related approaches were chosen. Especially it touched the Neural Network, because it used to predict only that particular one feature. The data were present in two columns, and various observations,depending on a used data sheet. Graphical representation is shown in fig. 4.1.

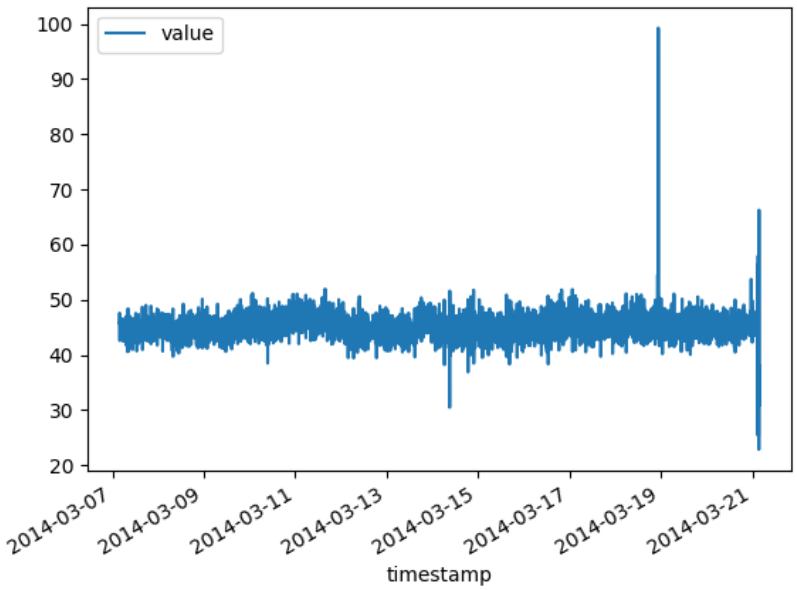

Fig. 4.1: Uni variate data graphical representation

There was another data set provided by a third person, describing behaviour of data warehouse during three days. It consists of thousands of features monitored for

every of three days, which are related to some indicators on particular machine. The format of the data is presented on fig. 4.3 and graphical representation in fig. 4.2.

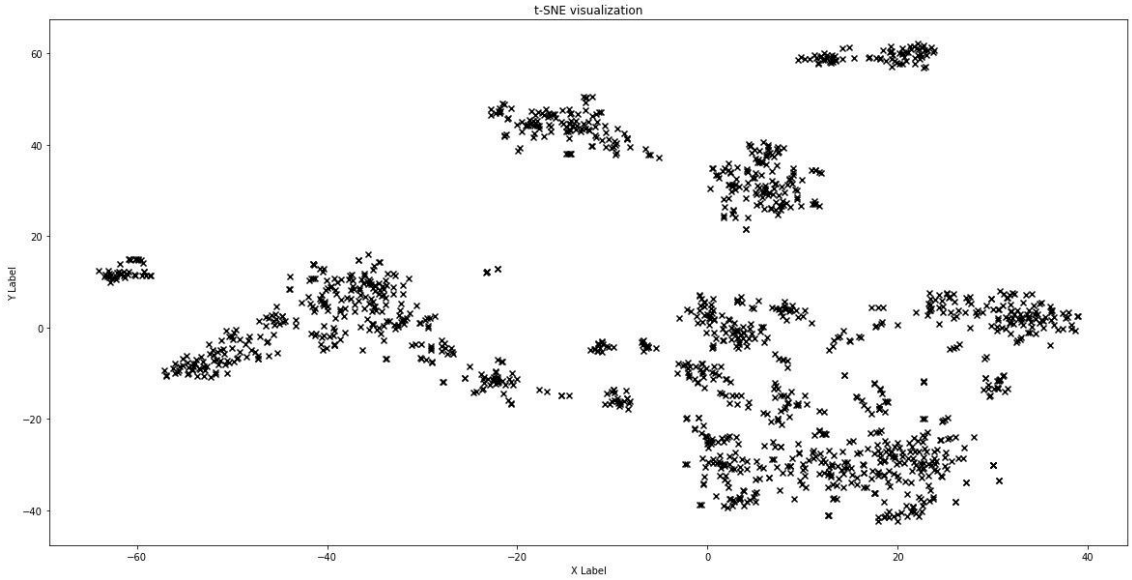

Fig. 4.2: DWH data graphical representation with t-SNE

	time	CL3-A.02(uxpvbwd0).0	CL3-A.02(uxpvbwd0).1	CL3-A.02(uxpvbwd0).2	CL3-A.03(uxpvr3d0).0	CL3-A.03(uxpvr3d0).1
300	2017-01-01 05:01:00	203	250	-3	-3	323
360	2017-01-01 06:01:00	100	153	-3	-3	190
361	2017-01-01 06:02:00	414	180	-3	-3	-3
376	2017-01-01 06:17:00	320	167	-3	-3	-3
377	2017-01-01 06:18:00	1078	105	-3	-3	-3
378	2017-01-01 06:19:00	124	140	-3	-3	1335
379	2017-01-01 06:20:00	445	114	-3	-3	-3

Fig. 4.3: DWH standardized logs

Also there was own data set created, which features were extracted based on open source KDD[24]. These data were collected within one month in one of data centers in Prague. Non Disclosure Agreement was signed with the company which was monitored, so it is unavailable provide full data set neither all information about the environment. The base is that mentioned data center is a central node for interconnection of more than 5 countries local data centers. The tested data set is consisting of normal behavior - one business day - and some of Cyber attacks as DDOS, SSH-Patator and Web Service brute force. In caption below the information

about the .csv file, used for the model fitting, is shown and in fig. 4.4 is initial table look presented.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975827 entries, 0 to 975826
Data columns (total 73 columns):
Destination Port                975827 non-null int64
 Flow Duration                  975827 non-null int64
 Total Fwd Packets              975827 non-null int64
 Total Backward Packets         975827 non-null int64
Total Length of Fwd Packets     975827 non-null int64
 Total Length of Bwd Packets    975827 non-null int64
 Fwd Packet Length Max          975827 non-null int64
 Fwd Packet Length Min          975827 non-null int64
 Fwd Packet Length Std          975827 non-null float64
```

Listing 4.1: Data set information

# Destina...	# Flow D...	# Total F...	# Total B...	# Total L...	# Total L...	# Fwd Pa...	# Fwd Pa...	# Fwd Pa...	# Fwd Pa...	# Bwd Pa...
88	609	7	4	484	414	233	0	69.14285714 29	111.9678950 584	207
88	879	9	4	656	3064	313	0	72.88888888 89	136.1538141 629	1532
88	1160	9	6	3134	3048	1552	0	348.2222222 222	682.4825598 097	1518
88	524	7	4	2812	2820	1397	0	401.7142857 143	679.9148756 243	1410
1034	6	1	1	6	6	6	6	6,00	0	6
88	1119	9	6	3160	3060	1565	0	351.1111111 111	688.2149817 543	1524
389	18378	13	13	4160	5724	1672	0	320,00	612.9740342 516	2634
88	822	7	4	458	356	220	0	65.42857142 86	105.6264757 575	178
88	876	9	4	630	2942	300	0	70,00	130.4223907 157	1471

Fig. 4.4: Present samples in data set

In all of the data sets anomalies have been included, that is why it has to be dividing this traffic to two parts: training and testing. All collected data are anonymized as one of the points for security insurance. The real data, that were captured from one company's datacenter, features have to be anonymized by using, for example, CityHash32 and additional operations: simple conversion to decimal format and then hashing. For example: IP address 127.0.0.1 is equal to 2130706433 and the hash is 2947144808. Hash function is any function that can be used for mapping original values, so-called "key" value, to a value of a certain length - the hash. For better data protection there cryptography hash functions are available, such SHA family. In that specific case of hash function the calculated output will be presented in hexadecimal format. To follow the data structure I would add the extra step of using non-cryptographic hash function to reduce the size of SHA function.

There is also a challenge to get data set from raw data captured, i.e. in .pcap format file. It is easy to export .pcap file as .csv using simple command for tshark

application. In this example, besides the export to file in .csv format, also specific extraction of features right from the packets is implemented.

```
tshark -r training_data.pcapng -T fields -e frame.time_epoch -e frame.
    len -e tcap.localValue -e tcap.result -E header=y -E separator=, -E
     occurrence=f -E quote=d > test.csv
```

Listing 4.2: Common tshark use for .pcap ⇒ .csv

Moreover, a correlation matrix has been generated to see, what attributes are correlating with each other.

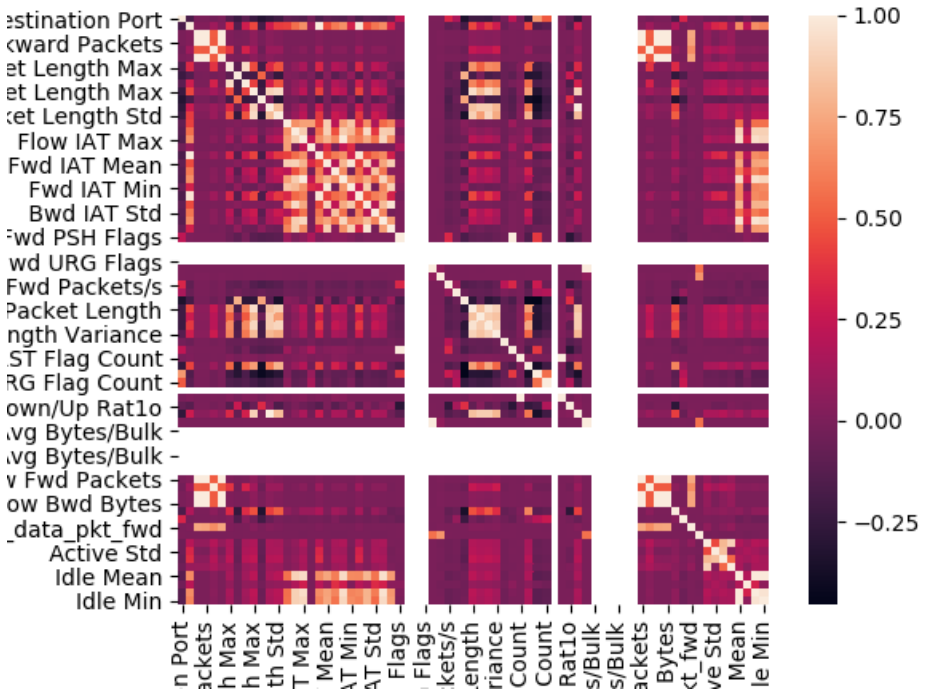

Fig. 4.5: Data correlation matrix

The problem of data flow to the module is that the solution should be consisting of several different detectors in the circuits, ideally. In addition, it is desirable that it will periodically launch an active scan to detect potential vulnerabilities, which is not yet implemented.

4.2 Environment description

As the run environment Kaggle Python kernel was used. Kaggle[1] is a platform for machine learning and data science competitions owned by Google,Inc. On this

[1] www.kaggle.com

platform users are able to find and share data sets, build models through the web interface, collaborate with other data scientists and machine learning engineers to solve published challenges. The common workflow for the Kaggle processes follows: a partner company publishes a competition with a set of data, specialists from all over the world are trying their best to solve the problem and after submitting, Kaggle automatically performs a cross validation on test data, that are hidden, and evaluates results for every committed solution. In the day of a deadline it stops to perform automatic tests and the leader becomes a winner. So, all the participants are using specially developed web-based kernels. Kaggle Kernels is a cloud computational environment which is accessible from any computer with only one condition - to have an account. There are few types of kernels presented in Kaggle: scripts, RMarkdown Scripts and Notebooks. Scripts are files that can be executed as code in a sequences. There are only R and Python programming languages supported. RMarkdown Scripts are scripts that executing RMarkdown code. From its name it's understood we have a mix of R and Markdown editing syntax. This combination is popular within the R using community of Kaggle. Notebooks is shortened from "Jupyter Notebooks". Jupyter Notebooks is a web application consisting of interactive cells either of R/Python code or a markdown as a text box.

4.2.1 Technical specifications

Kaggle Kernels provide every user with following resources: CPU Specifications

- 4 CPU cores
- 17 Gigabytes of RAM
- 6 hours execution time
- 5 Gigabytes of auto-saved disk space
- 16 Gigabytes of temporary, scratchpad disk space

GPU Specifications NVIDIA Tesla K80

- 2 CPU cores
- 14 Gigabytes of RAM

The whole environment is already pre-configured based on a programming language chosen. Python is a good decision for this project because of a bunch of different libraries for creating models based on Neural Networks and for its speed in such tasks as big data processing. For the models presented, a few of typical packages for machine learning/data science were used.They are presented here:

```
import numpy as np
import pandas as pd
import matplotlib
import seaborn as sns
```

```
import matplotlib.dates as md
from matplotlib import pyplot as plt
from sklearn import preprocessing
from sklearn.ensemble import IsolationForest
from sklearn.svm import OneClassSVM
from keras.layers.core import Dense, Activation, Dropout
from keras.layers.recurrent import LSTM
from keras.models import Sequential
```

Listing 4.3: Imported packages for model creation

4.3 Experiments and analysis

As the beginning of the field research, a native approaches were studied. With the pre-configured environment a simple Notebook was created on Kaggle Kernel platform. There are few of native models presented: One-Class SVM, Isolated forest, RNN, with some other possible approaches. Besides these ones, data classification and clustering were tested as well.

4.3.1 Classification Techniques

Sometimes the classification models are doing a good analysis for anomaly detection problems. But it is based on a specific conditions and their correlation. The main conditions are easy [19]

1. Labeled training data;
2. Anomalous and normal classes are balanced, at least 1:5;
3. Data is not auto correlated.

If all of them are met, there is no need to use anomaly detection techniques and algorithms as adaBoost or Random Forest can be used. Which actually were implemented during the Junction Hackathon[2] in Helsinki this year. The challenge was about finding anomalies based on attacks in Signaling System 7. Participants were given the raw data set in .pcap format, which was expected to be converted to .csv file. The final approach was based on an ensemble of few models. The main challenge here is not just to build up a model and fit it with the data, but the data itself. It was about determining what could be an anomaly and implement new features based on this human evaluation. The common $sklearn, pandas, numpy$ packages were used while building the model:

```
# Models
```

[2]https://projects.hackjunction.com/

```
clf1 = ensemble.RandomForestClassifier(random_state=rs, max_depth=8,
    n_jobs=-1)
clf2 = ensemble.AdaBoostClassifier(random_state=rs, n_estimators=100,
    learning_rate=0.1)
eclf = ensemble.VotingClassifier(estimators=[('rf', clf1), ('ada', clf2
    ),
                                 voting=voting_type, weights=weights)

# Fit the models
print("\nfitting..")
for clf in tqdm([clf1, clf2, eclf], total=3):
    clf.fit(X,y)
    clf.fit(X,y)
    clf.fit(X,y)

# Cross-validation evaluation
print("\nevaluating..")
for clf, label in tqdm(zip([clf1, clf2, eclf], \
                      ['Random Forest', 'AdaBoost',
                       'Voting Ensemble']), total=3):
    scores = cross_val_score(clf, X, y, cv=5, scoring='f1_macro',
    n_jobs=-1)
    print("f1-score: %0.2f (+/- %0.2f) [\%s]" % (scores.mean(), scores.
    std(), label))
```

Listing 4.4: Sample of classifiers code

The cross-validation was successful: all the anomalies were catched by the created model. As the conclusion, the second place in the international competition was gained.

4.3.2 Unsupervised techniques

One Class SVM and Isolation Forest

A Support Vector Machine is typically being associated with supervised learning, but there is one exception as One Class SVM. It can be used to identify anomalies, specifically collective ones. The algorithm clusters the normal data samples using training set, and after that, during the cross-validation, it tunes itself for anomaly detection the are the outside of the learned sequence.

```
0     962703
1      13124
Name: SVM, dtype: int64
```

Listing 4.5: Output of SVM for PRG data set: 13124 anomalous entries were found

In previous work, SVM was showing the best results out of all others, and here it shows good results as well, as in data set there are 13836 anomalous events recorded. While for Data Warehouse monitoring data set it has shown the same average performance:

```
0      1381
1        59
Name: SVM, dtype: int64
```

Listing 4.6: Output of SVM for DWH data set: 59 anomalous entries were found

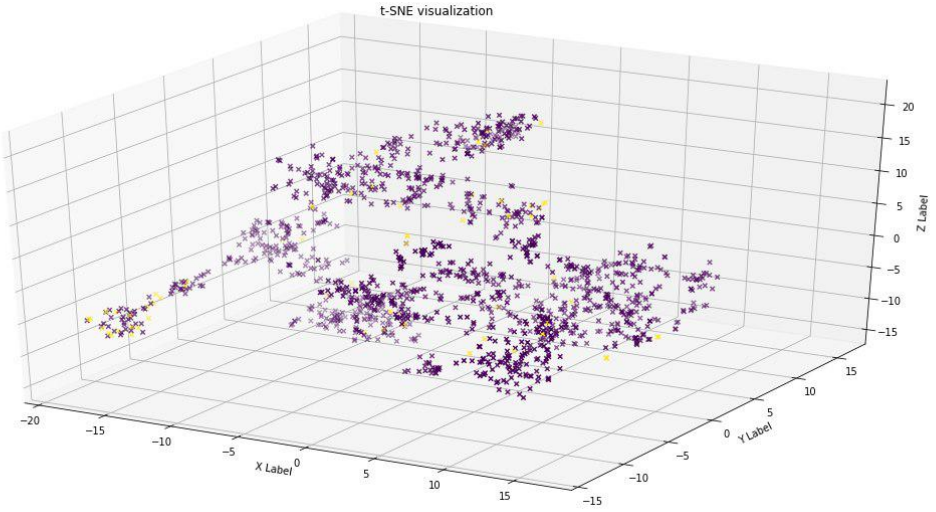

Fig. 4.6: SVM model anomaly detection in DWH data set with t-SNE. Yellow points are anomalous.

The result of uni variate data set anomaly detection by Support Vector Machine can be found on fig. 4.7

What is different with Isolation Forest, that Isolation Forest explicitly identifies anomalies instead of profiling normal data points. Isolation Forest, like any tree ensemble method, is built on the basis of decision trees. In these trees, partitions are created by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature.

```
0      966068
1        9759
Name: Iso, dtype: int64
```

Listing 4.7: Output of Isolation Forrest for PRG data set: 9759 anomalous entries were found

But from the experiment and out-of-the-box model, it is not quite right. As the first 529920 records were fully benign, and Isolation Forest have marked hundreds of

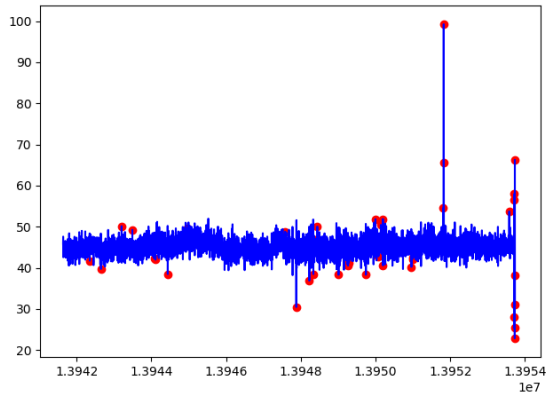

Fig. 4.7: SVM model anomaly detection in value targeted data set

records as anomaly, there is conclusion, that the lone model is not the best solution in such case. However, it should be noted, that the 80% of anomalous records have been detected in attack areas. The same behavior can be observed with the Prague data set:

```
0      1368
1        72
Name: Iso, dtype: int64
```

Listing 4.8: Output of Isolation Forest for DWH data set: 72 anomalous entries were found

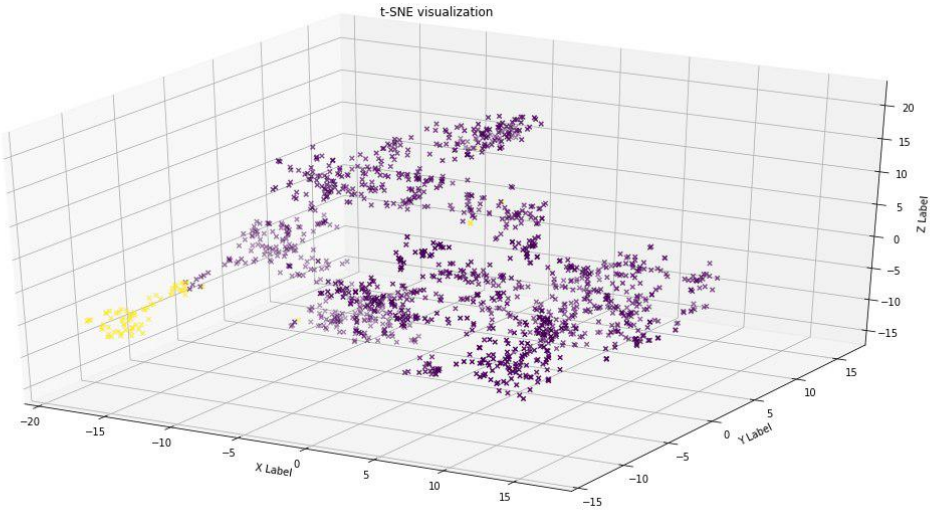

Fig. 4.8: Isolation Forest model anomaly detection in DWH data set with t-SNE. Yellow points are anomalous.

4.3.3 RNN

The LSTM RNN uses traffic time series and past values from the same season as input for the visible layer. Each such feature has a weight associated with it based on the hidden layer. The RNN's hyper-parameters are customizable. The modal is defined by the weights and at the end of the training phase, the model can be saved in memory. This LSTM model can ingest a window of traffic data, as past seasonal points, and can make a forecast of the upcoming traffic. If needed, the RNN gives room for improvement. One is able to inject new training data and actualize the model by building on top of the original one, effectively odding to the initial training. This ensures a forecast in the context of past, but also recent or new events. RNN makes predictions based on previous sequences. The anomaly is detected, when the data values are lying much further than the prediction of the model. For this experiments the sequences of 50 previous points are being learnt to predict 1 next value. The general architecture of network consisted of 3 layers with 150 units each, activation function was set to ReLU, as it enhances the learning tempo and it is able to avoid a problem of vanishing gradient. The run of solution was set as follows: first, the algorithm predicts the values of each of the observed indicators at time t, after which the values of these predictions are compared with the true values of these indicators at time t. Those objects, in which the difference between the predicted values and the true values exceeds a certain threshold are assumed as anomalous and being added to the main data set for representative purposes.

For the semestral project uni variate LSTM network was used. It was configured in the same way as general model, described above, excluding the set of prediction for every value of indicator. Meaning, that model was working only at prediction of one value.

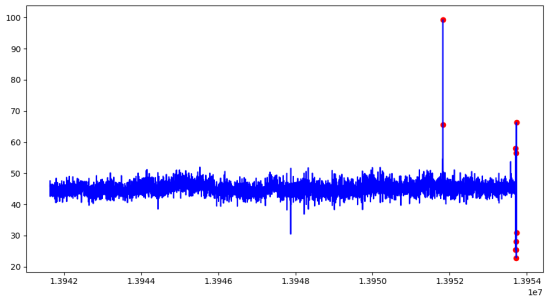

Fig. 4.9: NN with LSTM modules output for one dimensional data set

During the experiment with PRG data set, configured environment with 150 epochs took for over than 5 hours to end, but because of kernel online session is not able to be held for longer period of time, no data extraction from kernel is available.

Still, 12500 anomalies were found in cut data set, which means it has been tested right on set with anomalies.

For the experiment with DWH data set, LSTM net showed the worst result after all above described algorithms. It has detected a wide range of anomalous areas during one day. It is caused by very low number of data fed to the network as input. It is not able to construct all the dependencies for better reproduction based on such lack of information. It is able to see how many benign data point were marked as anomalous even with the same anomaly fraction set for every model constructed. See the table below for representation.

```
0.0     1176
1.0      239
Name: RNN, dtype: int64
```

Listing 4.9: Output of RNN LSTM for DWH data set: 239 anomalous entries were found

4.3.4 Evaluation

Based on presented results, Isolation Forest is showing better performance among all data sets presented. However, Neural Network proved itself better in cooperation with huge amount of data. While Support Vector Machine is considered as the oldest approach to solve such problems as anomaly detection, it still can stand at the same level with other native algorithms for Anomaly detection.

5 CONCLUSION AND FURTHER WORK

In this project the task of developing fully functioning IDS conceptual module is limited, instead, the module for Security Information and Event Management or Log management connection was created. As it was shown before, the data can be provided in a clear form, which means in the form of a set of functions that module can use in the input to the solution: as the result, the full pipeline has been applied. It is desirable to implement a multi-class method for presenting detector results at the end of the project, in order not only to detect the presence of anomalous activity, but also to determine its type, which could be a good continuity of this work. The native anomaly detection techniques such as One Class SVM, Isolation Forest and RNN LSTM were tested and showed an overall good results, which could be considered as successful fulfill of the primary task – an anomaly detection in multidimensional data. During the experiments three baseline models were built. Each of it was tested on two different types of data sets: uni variate and multivariate. All of the models showed very good output for the one value prediction and evaluation process, but all of the enterprise companies require to measure and process several to many values for each time stamp. That's why the redesign took a place in this thesis and was successful. While Isolation forest and Support Vector Machine were not in need of many changes, the RNN model had to be reconstructed. It leads to adding one more advantage to Isolation Forest method between others with its better adaptability. However, the Neural Networks are working better with big amounts of data, which makes them more suitable for processing log data from Data centers.

BIBLIOGRAPHY

[1] ACM Computing Surveys. New York: Association for Computing Machinery. ISSN 03600300.

[2] BOX, George E. P., Gwilym M. JENKINS a Gregory C. REINSEL. Time series analysis: forecasting and control. Fifth edition. Hoboken: John Wiley & Sons, 2016. Series in probability and statistics (Wiley). ISBN 1118675029.

[3] BUDUMA, Nikhil. Fundamentals of Deep Learning: Designing Next-Generation Artificial Intelligence Algorithms. Sebastopol: OReilly Media, 2016.

[4] CALDER, Alan a Steve WATKINS. International IT governance: an executive guide to ISO 17799/ISO 27001. Philadelphia, PA: Kogan Page Limited, 2006. ISBN 0749447486.

[5] Data Center Virtualization. Cisco Solutions [online]. 2019, 1 [cit. 2019-05-27]. Available from URL: <https://www.cisco.com/c/en/us/solutions/data-center-virtualization/index.html>.

[6] Deep AI [online]. Deep AI, 2017 [cit. 2018-12-10]. Available from URL: <https://deepai.org>.

[7] GAL, Yarin. Uncertainty in Deep Learning. Gonville and Caius College, 2016. PhD Dissertation. University of Cambridge.

[8] GUYON, Isabelle. Feature extraction: foundations and applications. Berlin: Springer-Verlag, 2006.

[9] HAN, Jiawei, KAMBER, Micheline and PEI, Jian. Data mining: concepts and techniques. Amsterdam: Elsevier/Morgan Kaufmann, 2012.

[10] HEBB, D. O.The Organization of Behavior: A Neuropsychological Theory, 2005, Taylor & Francis. ISBN 978-1-135-63190-1.

[11] LAGOUN, Sabria. Cracking the code: neuronal networks in the brain. WeAreDevelopers [online]. Vienna, 2018, 2018 [cit. 2018-12-5]. Available from URL: <https://www.wearedevelopers.com/ai-congress/videos>.

[12] LIU, Fei Tony, Kai Ming TING a Zhi-Hua ZHOU. Isolation-based Anomaly Detection. Data Mining [online]. Eighth IEEE International Conference on. — IEEE, 2008, 2008, 44 [cit. 2019-07-11].

[13] MAHMOOD, T. Security Analytics: Big Data Analytics for Cybersecurity: A Review of Trends, Techniques and Tools. 2nd National Conference on Information Assurance (NCIA), December 11, 2013, Rawalpindi, Pakistan.

[14] MARTÍN, C. A., J. M. TORRES, R. M. AGUILAR and S. DIAZ. Using Deep Learning to Predict Sentiments: Case Study in Tourism. Complexity. 2018, 2018(Volume 2018), 9. DOI: 7408431.

[15] MINASHKIN, V. G., R. A. SHMOYLOVA, N. A. SADOVNIKOVA, L. G. MOISEYKINA and Y. S. RYBAKOVA. Theory of Statistics. 4. Moscow: EAOI, 2008. ISBN 978-5-374-00041-2.

[16] Neural Designer [online]. Artificial Intelligence Techniques, 2018 [cit. 2018-12-13]. Available from URL: <https://www.neuraldesigner.com/>.

[17] NIELSEN, M. A., Neural Networks and Deep Learning [online]. Determination Press, 2015 [cit. 10. 12. 2018]. Available from URL: <http://neuralnetworksanddeeplearning.com/>.

[18] COPELAND, Michael. The Difference Between AI, Machine Learning, and Deep Learning? [online]. The Official NVIDIA Blog, 2017. Available from URL: <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>.

[19] PERERA, S. Introduction to Anomaly Detection: Concepts and Techniques. IwRinger [online]. 2008, 2018 [cit. 2018-10-17]. <https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/>.

[20] PRECUP, C. Network Predictive Analysis: A Journey into Traffic Forecasting with Deep Learning. Xrdocs.io [online]. 2018 [cit. 2019-05-27]. Available from URL: <https://xrdocs.io/telemetry/blogs/2018-07-19-network-predictive-analysis/>.

[21] RAJAHALME, J., A. CONTA, B. CARPENTER a S. DEERING. IPv6 Flow Label Specification. RFC 3697 [online]. Network Working Group: Standards Track, 2004, s. 9 [cit. 2019-05-27]. Available from URL: <https://www.ietf.org/rfc/rfc3697.txt>.

[22] ROUSE, M. Network traffic. SearchNetworking [online]. 2019, 2019(3), 1 [cit. 2019-05-27]. Available from URL: <https://searchnetworking.techtarget.com/definition/network-traffic>.

[23] SALZBERG, Steven L. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine Learning [online], 1994 DOI: 10.1007/BF00993309. ISSN 0885-6125. <http://link.springer.com/10.1007/BF00993309>.

[24] SHARAFALDIN I., Lashkari A. H., Ghorbani A. A., Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 4th International Conference on Information Systems Security and Privacy (ICISSP, Portugal, January 2018

[25] StatSoft [online]. Moscow: StatSoft, 2012 [cit. 2018-10-15]. Available from URL: <http://www.statsoft.ru/home/textbook/default.html/>.

[26] Understanding LSTM Networks. Colah's blog [online]. Github,Inc., 2018, 27.08.2015 [cit. 2018-11-14]. Available from URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

LIST OF FIGURES